

5

TITLE OF THE INVENTION  
PARALLEL DATA BUS WITH BIT POSITION ENCODED ON  
THE CLOCK WIRE

10

CROSS REFERENCE TO RELATED APPLICATIONS

This application claims priority of U.S. Provisional  
Patent Application No. 60/245,895 filed November 3, 2000  
entitled PARALLEL DATA BUS WITH BIT POSITION ENCODED ON  
15 THE CLOCK WIRE.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR  
DEVELOPMENT

N/A

20

BACKGROUND OF THE INVENTION

The present invention relates generally to high  
speed data transmission systems, and more specifically to  
a system and method for reliably transmitting parallel  
25 data over a plurality of high speed serial lines.

Data transmission systems are known that employ a  
plurality of serial lines for transmitting parallel data  
from a source to a destination. In a conventional data  
transmission system, parallel data to be transmitted is  
30 typically segregated into a plurality of narrower  
parallel data bytes or words. Next, the plurality of

-1-

parallel data bytes/words is serialized for transmission to the destination over a plurality of serial lines. At the destination, serial data streams carried by the respective lines are converted from serial to parallel  
5 form to reproduce the plurality of parallel data bytes/words, which are then aligned to regenerate the parallel data with its original ordering of data.

One drawback of the above-described data transmission system is that variations in, e.g., the  
10 lengths of the serial lines and/or the logic speeds associated with the serial lines can cause the serial data streams carried by the respective lines to be skewed. For example, corresponding data bits included in the serial data streams may arrive at the destination  
15 during different clock periods. This can be particularly problematic for high speed data transmission systems employing serial data transmission rates on the order of, e.g., 2.5 GHz, which may require corresponding serialized data bits to arrive during the same 400 psec clock  
20 period. Such data skew can make it difficult to align the data received at the destination and regain the original ordering of the transmitted parallel data.

Various encoding techniques have been developed to address, at least in part, the problem of data skew in  
25 the transmission of data over high speed serial lines. One such encoding technique, commonly known as the 8B/10B data transmission code, segregates the parallel data to be transmitted into a plurality of parallel data bytes, and encodes the parallel data bytes to form corresponding  
30 10-bit parallel data words, which are then serialized for

transmission to the destination over respective lines. Each 10-bit parallel data word is typically encoded to include alignment information, which is used at the destination for properly aligning the parallel data  
5 despite the occurrence of data skew. However, the 8B/10B data encoding technique also has drawbacks. For example, because the wider 10-bit parallel data words are serialized for transmission to the destination over the serial lines rather than the narrower parallel data  
10 bytes, the serial data transmission rate is frequently increased to achieve a desired level of performance.

It would therefore be desirable to have an improved system and method for transmitting parallel data from a source to a destination over a plurality of high speed  
15 serial lines. Such a high speed data transmission system would be capable of reliably transmitting parallel data to the destination despite the occurrence of data skew. It would also be desirable to have a high speed data transmission system that can reliably transmit parallel  
20 data without requiring an increase in the serial data transmission rate.

#### BRIEF SUMMARY OF THE INVENTION

In accordance with the present invention, a system  
25 and method for transmitting parallel data from a source to a destination over a plurality of high speed serial lines is provided that operates reliably even in the presence of data skew. Benefits of the presently disclosed invention are achieved by encoding alignment  
30 information for the parallel data on a clock transmitted

to the destination over one of the high speed serial lines.

In one embodiment, the high speed data transmission system includes a protocol generator, a de-skew circuit, and a plurality of high speed serial lines coupled between the protocol generator and the de-skew circuit. Serial data streams are transmitted over the plurality of high speed serial lines at a predetermined serial data transmission rate. The protocol generator, which operates at a suitable fraction of the predetermined serial data transmission rate, is configured to input information from a wide bus and output information to a plurality of narrower buses. In the presently disclosed embodiment, the predetermined serial data transmission rate is equal to 2.5 GHz, and the protocol generator inputs information from a single 128-bit bus at 311 MHz and outputs information to seventeen (17) 8-bit buses at 311 MHz.

The information that is inputted by the protocol generator from the wide bus comprises parallel data to be transmitted to the destination. Further, the information that is outputted by the protocol generator to the narrower buses comprises a plurality of parallel bytes conforming to a predetermined protocol. One of the parallel bytes is used to generate a clock, while the remaining parallel bytes comprise the parallel data to be transmitted to the destination. The parallel byte for generating the clock has alignment information encoded thereon, which is subsequently used for properly aligning the parallel data bytes at the destination to regain the

original data ordering of the parallel data. In the presently disclosed embodiment, the alignment information encoded on the clock comprises at least one data bit of each parallel data byte.

5           The information that is outputted by the protocol generator to the plurality of narrow buses is serialized before being transmitted to the destination over the plurality of high speed serial lines. At the destination, the serial data streams carried by the  
10       respective lines are converted from serial to parallel form to reproduce the plurality of parallel bytes.

          The de-skew circuit, which also operates at a suitable fraction of the predetermined serial data transmission rate, is configured to input the plurality  
15       of reproduced parallel bytes from a plurality of narrow buses and output parallel data comprising the parallel data bytes to a wider bus. In the disclosed embodiment, the de-skew circuit inputs the parallel bytes from  
20       seventeen (17) 8-bit buses at 311 MHz and outputs the parallel data to a single 128-bit bus at 311 MHz. One of the inputted parallel bytes is derived from the clock, and the remaining sixteen (16) parallel bytes comprise the transmitted parallel data. The parallel data  
25       outputted by the de-skew circuit has the same data ordering as the parallel data originally inputted by the protocol generator.

          The de-skew circuit is further configured to use the alignment information encoded on the bytes derived from the clock for properly aligning the parallel data bytes  
30       before outputting the parallel data over the wide bus.

In the disclosed embodiment, the predetermined protocol requires that the alignment information encoded on the clock include a single bit from each of the parallel data bytes. Specifically, the alignment information includes  
5 the Most Significant Bit (MSB) of the upper nibble of a first parallel data byte, and the MSB of the lower nibble of a next contiguous parallel data byte. The alignment information then alternates between including the MSB of the upper nibble and the MSB of the lower nibble of  
10 subsequent contiguous parallel data bytes until a single bit from each of the sixteen (16) parallel data bytes is encoded on the clock.

The de-skew circuit selects respective bit positions in the bytes derived from the clock and the first  
15 parallel data byte, and compares the bits in the selected bit positions a predetermined number of times. In the event the de-skew circuit detects no mismatches, it is concluded that the position of the single bit from the first parallel data byte included in the alignment  
20 information is located in the same bit position in both the bytes derived from the clock and the first parallel data byte. The above-described steps are then repeated for a next contiguous parallel data byte.

In the event the de-skew circuit detects a mismatch,  
25 the de-skew circuit selects another bit position in the bytes derived from the clock and/or the first parallel data byte and repeats the above-described comparison(s). In the event the de-skew circuit successively selects each bit position in the bytes derived from the clock  
30 and/or the first parallel data byte and detects a

mismatch for each bit position, it is concluded that a bit error has occurred on one of the serial lines. The above-described steps may then be repeated for the first parallel data byte.

5           In the event the de-skew circuit repeats the above-described steps and detects no mismatches for the next contiguous parallel data byte, the relative bit positions of the first and the next contiguous parallel data bytes are determined at the destination and these contiguous  
10 data bytes are then aligned. In the disclosed embodiment, the de-skew circuit aligns the contiguous data bytes by temporarily storing the data bytes in a memory or buffer with the data bits in their original order.

15           The above-described steps are then repeated for each remaining parallel data byte. In the event the de-skew circuit detects no mismatches for the remaining parallel data bytes, the relative bit positions of the sixteen (16) contiguous parallel data bytes are determined and  
20 the contiguous data bytes are properly aligned. Finally, the de-skew circuit outputs the parallel data comprising the de-skewed parallel data bytes over the wide bus with the original ordering of data restored.

25           By encoding alignment information, i.e., data bit positions, on a clock transmitted with parallel data to a destination over a plurality of high speed serial lines, the transmitted parallel data can be de-skewed at the destination to regain the original ordering of the data. Because the parallel data is not encoded to include  
30 additional bits before being serialized and transmitted

to the destination (as in, e.g., the 8B/10B data encoding technique), the serial data transmission rate need not be increased to achieve a desired performance level.

Other features, functions, and aspects of the invention will be evident from the Detailed Description of the Invention that follows.

#### BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

The invention will be more fully understood with reference to the following Detailed Description of the Invention in conjunction with the drawings of which:

Fig. 1 is a block diagram depicting a high speed data transmission system according to the present invention;

Fig. 2 is a block diagram depicting a protocol generator included in the high speed data transmission system of Fig. 1;

Fig. 3 is a block diagram depicting a de-skew circuit included in the high speed data transmission system of Fig. 1; and

Fig. 4 is a timing diagram depicting a bus protocol employed by the high speed data transmission system of Fig. 1.

#### DETAILED DESCRIPTION OF THE INVENTION

U.S. Provisional Patent Application No. 60/245,895 filed November 3, 2000 is incorporated herein by reference.

A system and method for transmitting parallel data from a source to a destination over a plurality of high



speed serial lines is disclosed that operates reliably despite the occurrence of data skew. Such reliable operation is achieved by encoding alignment information relating to selected bit positions of the parallel data on a clock transmitted to the destination over one of the high speed serial lines, and using the alignment information at the destination to regain the original ordering of the data.

Fig. 1 depicts an illustrative embodiment of a high speed data transmission system 100, in accordance with the present invention. In the illustrated embodiment, the high speed data transmission system 100 includes a protocol generator 102, a de-skew circuit 112, and a plurality of high speed serial lines 120.0-120.15 and 122 coupled between the protocol generator 102 and the de-skew circuit 112. The protocol generator 102 is configured to input parallel data PG\_DIN from a wide bus 114, and output segregated parallel data PG\_D0-PG\_D15 over a plurality of narrower buses 116.0-116.15 and clock protocol data PG\_P over a narrower bus 118.

The data transmission system 100 further includes a plurality of Parallel/Serial (P/S) converters 104.0-104.15 configured to convert the segregated parallel data PG\_D0-PG\_D15 to serial data D0-D15, respectively, for transmission over the plurality of high speed serial lines 120.0-120.15; and, a P/S converter 106 configured to convert the clock protocol data PG\_P to a clock CLK having a predetermined clock rate for transmission over the high speed serial line 122. The P/S converter 106 is further configured to generate a clock PG\_CLK on a line

128 for use by the protocol generator 102 in generating the parallel data PG\_D0-PG\_D15 and the clock protocol data PG\_P. In the illustrated embodiment, the PG\_CLK clock rate is a suitable fraction of the CLK clock rate.

5           Moreover, the data transmission system 100 includes a plurality of Serial/Parallel (S/P) converters 108.0-108.15 configured to convert the serialized data D0-D15 to parallel data DS\_D0-DS\_D15, respectively; and, an S/P converter 110 configured to convert the clock CLK to  
10 clock protocol data DS\_P. The S/P converter 110 is further configured to recover clocks DS\_CLK\_0-DS\_CLK\_15 for use by the S/P converters 108.0-108.15, respectively, and a clock DS\_CLK for use by the de-skew circuit 112, from the clock CLK. In the illustrated embodiment, the  
15 clock rate of each clock DS\_CLK\_0-DS\_CLK\_15 is equal to the CLK clock rate, and the DS\_CLK clock rate is a suitable fraction of the CLK clock rate. The P/S converters 104.0-104.15 and 106 and the S/P converters 108.0-108.15 and 110 may comprise conventional circuitry  
20 for recovering clocks and serial/parallel data.

The de-skew circuit 112 is configured to input the parallel data DS\_D0-DS\_D15 via a plurality of narrow buses 124.0-124.15, respectively, and the clock protocol data DS\_P via a narrow bus 126; and, output parallel data  
25 DS\_DOUT over a wider bus 134. The de-skew circuit 112 uses the clock protocol data DS\_P for properly aligning the parallel data DS\_D0-DS\_D15. Further, the de-skew circuit 112 inputs control values ERR\_CMP and SAM\_CMP, and outputs a control signal LOCK, the functions of which  
30 are described below.

The parallel data DS\_D0-DS\_D15 essentially comprises a reproduction of the parallel data PG\_D0-PG\_D15, respectively, and the clock protocol data DS\_P essentially comprises a reproduction of the clock protocol data PG\_P. It is noted, however, that the relative bit positions of the parallel data DS\_D0-DS\_D15 may be skewed as a result of the data transmission from the protocol generator 102 to the de-skew circuit 112 over the high speed serial lines 120.0-120.15.

10 In the illustrated embodiment, the plurality of serial lines 120.0-120.15 is configured to run at about 2.5 GHz. Similarly, the clock CLK on the serial line 122 has a clock rate of about 2.5 GHz, and the clocks DS\_CLK\_0-DS\_CLK\_15 recovered by the S/P converter 110 have clock rates of about 2.5 GHz. Further, the clock PG\_CLK generated by the P/S converter 106 and the clock DS\_CLK recovered by the S/P converter 110 have clock rates of one-eighth the CLK clock rate or about 311 MHz. Moreover, each of the buses 114 and 134 is 128 bits wide, and each of the buses 116.0-116.15, 118, 124.0-124.15, and 126 is 8 bits wide. It should be understood, however, that in alternative embodiments, the data transmission system 100 may be configured to comprise wider or narrower buses running at higher or lower clock rates.

25 It should be further understood that the functions of the data transmission system 100 described herein may be software-driven and executable out of a memory by a processor, embodied in part or in whole using hardware components such as custom or semi-custom integrated

circuits such as Application Specific Integrated Circuits (ASICs), controllers, or other hardware components or devices, or a combination of hardware components and software. In the illustrated embodiment, the protocol  
5 generator 102 and the de-skew circuit 112 are embodied in one or more CMOS ASICs.

Fig. 2 depicts an illustrative embodiment of the protocol generator 102 included in the high speed data transmission system 100 (Fig. 1). In the illustrated  
10 embodiment, the protocol generator 102 inputs the parallel data PG\_DIN carried by the bus 114, and outputs the segregated parallel data PG\_D0-PG\_D15 over the plurality of buses 116.0-116.15 and the clock protocol data PG\_P over the bus 118, in accordance with a  
15 predetermined bus protocol.

Fig. 4 depicts an exemplary bus protocol employed by the protocol generator 102 for outputting the parallel data PG\_D0-PG\_D15 and the clock protocol data PG\_P. As described above, the serial data D0-D15 and the clock CLK  
20 are derived from the parallel data PG\_D0-PG\_D15 and the clock protocol data PG\_P, respectively. It is noted that Fig. 4 omits an explicit depiction of the bus protocol for the serial data D0-D11 for clarity of discussion. It is further noted that Fig. 4 depicts bit positions of the serial data D12-D15 relative to the clock CLK with no  
25 skew among the data bits.

As mentioned above, each of the buses 116.0-116.15 and 118 is disclosed as being 8 bits wide. Fig. 4 therefore depicts the relative positions of bits 15\_7-  
30 15\_0 of serial data D15, bits 14\_7-14\_0 of serial data

D14, bits 13\_7-13\_0 of serial data D13, and bits 12\_7-12\_0 of serial data D12. Specifically, in the event there is no data skew, bits 15\_7-15\_0 and bits 14\_7-14\_0 are asserted during a time interval T<sub>0</sub>-T<sub>7</sub>, and bits 13\_7-13\_0 and bits 12\_7-12\_0 are asserted during a time interval T<sub>8</sub>-T<sub>15</sub>. It follows that bits 11\_7-11\_0 of serial data D11 and bits 10\_7-10\_0 of serial data D10 are asserted during a time interval T<sub>16</sub>-T<sub>23</sub>, bits 9\_7-9\_0 of serial data D9 and bits 8\_7-8\_0 of serial data D8 are asserted during a time interval T<sub>24</sub>-T<sub>31</sub>, bits 7\_7-7\_0 of serial data D7 and bits 6\_7-6\_0 of serial data D6 are asserted during a time interval T<sub>32</sub>-T<sub>39</sub>, bits 5\_7-5\_0 of serial data D5 and bits 4\_7-4\_0 of serial data D4 are asserted during a time interval T<sub>40</sub>-T<sub>47</sub>, bits 3\_7-3\_0 of serial data D3 and bits 2\_7-2\_0 of serial data D2 are asserted during a time interval T<sub>48</sub>-T<sub>55</sub>, and bits 1\_7-1\_0 of serial data D1 and bits 0\_7-0\_0 of serial data D0 are asserted during a time interval T<sub>56</sub>-T<sub>63</sub>.

In the illustrated embodiment, alignment information comprising a single data bit from each of the serial data D0-D15 is included on the clock CLK. As shown in Fig. 4, bit 15\_7 of serial data D15, bit 14\_3 of serial data D14, bit 13\_7 of serial data D13, and bit 12\_3 of serial data D12 are included on the clock CLK. It follows that bit 11\_7 of serial data D11, bit 10\_3 of serial data D10, bit 9\_7 of serial data D9, bit 8\_3 of serial data D8, bit 7\_7 of serial data D7, bit 6\_3 of serial data D6, bit 5\_7 of serial data D5, bit 4\_3 of serial data D4, bit 3\_7 of serial data D3, bit 2\_3 of serial data D2, bit 1\_7 of serial data D1, and bit 0\_3 of serial data D0 are also

included on the clock CLK. It is noted that the relative positions of the data bits 15\_7, 14\_3, 13\_7, 12\_3, 11\_7, 10\_3, 9\_7, 8\_3, 7\_7, 6\_3, 5\_7, 4\_3, 3\_7, 2\_3, 1\_7, and 0\_3 on the clock CLK are indicative of the relative positions of these bits in the serial data D15-D0 with no data skew.

It is further noted that, in accordance with the presently disclosed bus protocol, the clock CLK includes the bit 15\_7 during time interval  $T_0$  and inverted versions of the bit 15\_7 (shown as "~15\_7") during time intervals  $T_1$ - $T_3$ . The clock CLK similarly includes inverted bits ~14\_3, ~13\_7, ~12\_3, ~11\_7, ~10\_3, ~9\_7, ~8\_3, ~7\_7, ~6\_3, ~5\_7, ~4\_3, ~3\_7, ~2\_3, ~1\_7, and ~0\_3 during the three time intervals immediately following the respective assertions of these bits without inversion. In this way, it is assured that the edge density of the clock CLK is sufficient to allow the S/P converter 110 to recover the clocks DS\_CLK\_0-DS\_CLK\_15 from the clock CLK.

It should be understood that alternative bus protocols may be employed in which alignment information comprising one or more data bits from each of the serial data D0-D15 are included on the clock CLK, so long as the clock CLK has sufficient edge density to allow recovery of the clocks DS\_CLK\_0-DS\_CLK\_15.

As shown in Fig. 2, the protocol generator 102 includes a parallel data segregator 240 and a clock protocol data generator 242. The parallel data segregator 240 is configured to input the parallel data PG\_DIN at the PG\_CLK clock rate, and output the segregated parallel data PG\_D0-PG\_D15 at the PG\_CLK clock

rate so that the serial data D0-D15 derived therefrom conforms to the bus protocol depicted in Fig. 4. Similarly, the clock protocol data generator 242 is configured to input the parallel data PG\_DIN at the PG\_CLK clock rate, and output the clock protocol data PG\_P at the PG\_CLK clock rate so that the clock CLK derived therefrom conforms to the bus protocol of Fig. 4.

Specifically, the clock protocol data generator 242 includes a first multiplexor (MUX) 244, a second MUX 248, and a counter 246 clocked by the clock PG\_CLK and operatively connected to respective control terminals of the MUXs 244 and 248. Each of the MUXs 244 and 248 is configured to input the parallel data PG\_DIN. Further, the counter 246 is configured such that each tick of the clock PG\_CLK advances the counter 246, which applies suitable control signals to the respective MUX control terminals to allow the MUXs 244 and 248 to successively select different pairs of data bits from the parallel data PG\_DIN. For example, the MUX 244 may be controlled to select the data bit 15\_7 and the MUX 248 may be simultaneously controlled to select the data bit 14\_3. The MUX 244 may then provide the bit 15\_7 directly to a buffer 254, and provide three (3) inverted bits ~15\_7 to the buffer 254 via an inverter 250. Similarly, the MUX 248 may provide the bit 14\_3 directly to the buffer 254, and provide three (3) inverted bits ~14\_3 to the buffer 254 via an inverter 252. As a result, the buffer 254 includes the data bits 15\_7, ~15\_7, ~15\_7, ~15\_7, 14\_3, ~14\_3, ~14\_3, and ~14\_3, preferably in eight (8) contiguous locations.

Next, the buffer 254 outputs these 8 bits of clock protocol data PG\_P over the bus 118 for subsequent serialization and transmission over the serial line 122 as a portion of the clock CLK. The clock protocol data generator 242 successively processes the data bit pairs 13\_7 and 12\_3, 11\_7 and 10\_3, 9\_7 and 8\_3, 7\_7 and 6\_3, 5\_7 and 4\_3, 3\_7 and 2\_3, and 1\_7 and 0\_3 in a similar manner.

In a preferred embodiment, the parallel data PG\_DIN carried by the 128-bit bus 114 maps to the segregated parallel data PG\_D0-PG\_D15 carried by the 8-bit buses 116.0-116.15 as follows.

	PG_DIN[127:124,63:60] = PG_D15
	PG_DIN[123:120,59:56] = PG_D14
15	PG_DIN[119:116,55:52] = PG_D13
	PG_DIN[115:112,51:48] = PG_D12
	PG_DIN[111:108,47:44] = PG_D11
	PG_DIN[107:104,43:40] = PG_D10
	PG_DIN[103:100,39:36] = PG_D9
20	PG_DIN[99:96,35:32] = PG_D8
	PG_DIN[95:92,31:28] = PG_D7
	PG_DIN[91:88,27:24] = PG_D6
	PG_DIN[87:84,23:20] = PG_D5
	PG_DIN[83:80,19:16] = PG_D4
25	PG_DIN[79:76,15:12] = PG_D3
	PG_DIN[75:72,11:8] = PG_D2
	PG_DIN[71:68,7:4] = PG_D1
	PG_DIN[67:64,3:0] = PG_D0

Further, every eight (8) consecutive ticks of the clock PG\_CLK, the parallel data PG\_DIN carried by the



128-bit bus 114 successively maps to the clock protocol data PG\_P carried by the 8-bit bus 118 as follows.

PG\_DIN[127,~127,~127,~127,59,~59,~59,~59] = PG\_P  
PG\_DIN[119,~119,~119,~119,51,~51,~51,~51] = PG\_P  
5 PG\_DIN[111,~111,~111,~111,43,~43,~43,~43] = PG\_P  
PG\_DIN[103,~103,~103,~103,35,~35,~35,~35] = PG\_P  
PG\_DIN[95,~95,~95,~95,27,~27,~27,~27] = PG\_P  
PG\_DIN[87,~87,~87,~87,19,~19,~19,~19] = PG\_P  
PG\_DIN[79,~79,~79,~79,11,~11,~11,~11] = PG\_P  
10 PG\_DIN[71,~71,~71,~71,3,~3,~3,~3] = PG\_P

Fig. 3 depicts an illustrative embodiment of the de-skew circuit 112 included in the high speed data transmission system 100 (see Fig. 1). In the illustrated embodiment, the de-skew circuit 112 inputs the parallel data DS\_D0-DS\_D15 via the respective buses 124.0-124.15 and the clock protocol data DS\_P via the bus 126, and outputs the parallel data DS\_DOUT over the bus 134. It is noted that the parallel data DS\_D0-DS\_D15 is derived from the serial data D0-D15, respectively, and the clock protocol data DS\_P is derived from the clock CLK. Further, the parallel data DS\_DOUT outputted by the de-skew circuit 112 over the bus 134 has the same data ordering as the parallel data PG\_DIN originally inputted by the protocol generator 102.

25 The de-skew circuit 112 uses the clock protocol data DS\_P for properly aligning the parallel data DS\_D0-DS\_D15 to regain the original data ordering of the parallel data DS\_DOUT. Specifically, a plurality of First-In First-Out (FIFO) buffers 364.0-364.15 receives the parallel data  
30 DS\_D0-DS\_D15 over the respective buses 124.0-124.15, and

5 a buffer 366 receives the clock protocol data DS\_P over the bus 126. Next, the de-skew circuit 112 compares bit values in selected bit positions of the respective FIFO buffers 364.0-364.15 to bit values in selected bit positions of the buffer 366 to determine the relative bit positions of the parallel data DS\_D0-DS\_D15. The de-skew circuit 112 then uses this information relating to the relative bit positions to align the parallel data DS\_D0-DS\_D15 for subsequent output over the bus 134 as the parallel data DS\_DOUT.

10 The manner in which the de-skew circuit 112 determines the relative bit positions of the parallel data DS\_D0-DS\_D15 will be better understood with reference to the following illustrative example, in which the Most Significant Bit (MSB) of the parallel data DS\_D15 is located using the alignment information for DS\_D15 encoded on the clock protocol data DS\_P. First, a de-skew controller 368 applies a first control signal to the FIFO buffer 364.15 via a bit position selection circuit (POS) 360.15, and a second control signal to a MUX 370 via a data selection circuit (SEL) 372. The FIFO buffer 364.15 then serially provides the data DS\_D15 to the MUX 370.

20 In the illustrated embodiment, the FIFO buffer 364.15 is configured to accommodate up to 4 bit times of skew (about 1.6 nsecs) between the first and last arriving parallel data byte DS\_D0-DS\_D15. The FIFO buffer 364.15 may therefore be configured to store at least 12 data bits. Further, the first control signal applied to the FIFO buffer 364.15 via POS 360.15 may

cause the FIFO buffer 364.15 to serially provide 8 data bits to the MUX 370 starting with the data bit in the first bit position ("Bit 7") and continuing with the data bits in the next 7 consecutive bit positions (bits 6-0) of the FIFO buffer 364.15. Moreover, the second control signal applied to the MUX 370 via SEL 372 causes the MUX 370 to provide Bit 7 to an exclusive-or (XOR) gate 374. It is noted that the second control signal also causes the MUX 370 to provide bit 3 ("Bit 3") to an XOR gate 376.

Because of the possible occurrence of data skew in the transmission of the serial data D0-D15 over the serial lines 120.0-120.15, it is uncertain whether Bit 7 corresponds to the actual MSB of the data DS\_D15 (i.e., bit 15\_7). For this reason, the de-skew controller 368 further applies a third control signal to the buffer 366 via POS 362. For example, POS 362 may cause the buffer 366 to provide the data bit in the MSB position of DS\_P to the XOR gate 374. In this example, the data bit in the MSB position of DS\_P corresponds to the bit 15\_7 encoded on the clock CLK at time interval T<sub>0</sub> (see Fig. 4). It is noted that POS 362 also causes the buffer 366 to provide the data bit 14\_3 (Bit 3) encoded on the clock CLK at time interval T<sub>4</sub> to the XOR gate 376 according to the exemplary bus protocol depicted in Fig. 4.

As a result, the XOR gate 374 compares Bit 7 of the parallel data DS\_D15 to the corresponding Bit 7 of the clock protocol data DS\_P. In the event both of the values of these bits are either logical high or logical low, the XOR gate 374 outputs a logical low level. In

the event these bits have different values, the XOR gate 374 outputs a logical high level. In alternative embodiments, the XOR gate 374 may compare Bit 7 of DS\_D15 to the corresponding Bit 7 of DS\_P, and the XOR gate 376 may compare Bit 3 of DS\_D14 to the corresponding Bit 3 of DS\_P, simultaneously.

In the illustrated embodiment, Bit 7 of the parallel data DS\_D15 is compared to the corresponding Bit 7 of the clock protocol data DS\_P by the XOR gate 374 a desired number of times, as determined by the value SAM\_CMP. For example, the de-skew controller 368 may store a value in a cycle value register (CYC\_VAL) 384. Further, a cycle counter (CYC\_CTR) 386 may be configured to count repeatedly from 0 to the stored cycle value. Because the comparison of Bit 7 of the parallel data DS\_D15 to the corresponding Bit 7 of the clock protocol data DS\_P occurs only once every 64 ticks in the disclosed embodiment, the stored cycle value equals 64.

In the event a comparator (CMP) 382 detects that the output of CYC\_CTR 386 equals the value stored in the CYC\_VAL 384, the CMP 382 provides a logical high level (SAM) to an AND gate 378, thereby causing the AND gate 378 to pass the output of the XOR gate 374 to the de-skew controller 368 as a first error signal, ERR\_Bit7. It is noted that the logical high SAM level also causes an AND gate 380 to pass the output of the XOR gate 376 to the de-skew controller 368 as a second error signal, ERR\_Bit3. The de-skew controller 368 includes a sample counter (not shown) that counts the number of times that SAM is asserted.

In the event the sample counter reaches the value SAM\_CMP without ERR\_Bit7 being asserted, it is concluded that Bit 7 provided by the MUX 370 to the XOR gate 374 corresponds to the actual MSB of the parallel data DS\_D15, i.e., bit 15\_7. It is noted that while locating bit 15\_7 of the data DS\_D15, the second error signal, ERR\_Bit3, may be ignored. In the event ERR\_Bit7 is asserted before the sample counter reaches the value SAM\_CMP, another first control signal is applied to the FIFO buffer 364.15, which may cause the FIFO buffer 364.15 to serially provide 8 data bits to the MUX 370 starting with the bit in the second bit position and continuing with the bits in the next 7 consecutive bit positions of the FIFO buffer 364.15, thereby sliding the FIFO buffer output by one bit.

Further, another third control signal may be applied to POS 362 to cause the buffer 366 to provide a different pair of data bits as Bit 7 and Bit 3 to the XOR gates 374 and 376, respectively. For example, the buffer 366 may provide the data bit in the bit position of DS\_P corresponding to the bit ~15\_7 encoded on the clock CLK at time interval  $T_1$  (see Fig. 4). It is noted that POS 362 may also cause the buffer 366 to provide the data bit ~14\_3 encoded on the clock CLK at time interval  $T_3$  according to the exemplary bus protocol depicted in Fig. 4. In the event all possible combinations of data bits stored in the FIFO buffer 364.15 and the buffer 366 are compared and ERR\_Bit7 is asserted for each possible combination, it is concluded that a bit error has occurred on the serial line 120.15 (see Fig. 1). The

above-described steps for locating the actual MSB of the parallel data DS\_D15 may then be repeated.

Steps analogous to those described above for locating bit 15\_7 of the parallel data DS\_D15 using the alignment information encoded on the clock protocol data DS\_P may be performed to locate bit 14\_3, bit 13\_7, bit 12\_3, bit 11\_7, bit 10\_3, bit 9\_7, bit 8\_3, bit 7\_7, bit 6\_3, bit 5\_7, bit 4\_3, bit 3\_7, bit 2\_3, bit 1\_7, and bit 0\_3 of the parallel data DS\_D14, DS\_D13, DS\_D12, DS\_D11, DS\_D10, DS\_D9, DS\_D8, DS\_D7, DS\_D6, DS\_D5, DS\_D4, DS\_D3, DS\_D2, DS\_D1, and DS\_D0, respectively. In the event all possible combinations of data bits stored in the FIFO buffer corresponding to any one of the data DS\_D14-DS\_D0 and the data bits stored in the buffer 366 are compared and ERR\_Bit7 (or ERR\_Bit3) is asserted for each possible combination, it is concluded that the data bit of least one previous parallel data byte was incorrectly located and the above-described steps are repeated from the start, e.g., starting with the parallel data DS\_D15.

Once the single bits (Bits 7 and 3) of each pair of contiguous parallel data bytes are located, the relative bit positions of the pair of data bytes are known and the data byte pair can be properly aligned. In the illustrated embodiment, each contiguous pair of the parallel data bytes DS\_D0-DS\_D15 is aligned by temporarily storing the data bytes in a buffer 388 with the data bits of the data byte pair in their original order. Next, the de-skew controller 368 asserts the control signal LOCK, and the buffer 388 outputs the aligned parallel data DS\_D0-DS\_D15 over the bus 134 as

the parallel data DS\_DOUT such that the data DS\_DOUT has the same data ordering as the parallel data PG\_DIN originally inputted by the protocol generator 102.

5 In the disclosed embodiment, while the control signal LOCK is asserted, the de-skew circuit 112 continues to compare a single bit (Bit 7 or Bit 3) of each incoming parallel data byte DS\_D15-DS\_D0 with the corresponding alignment information encoded on the clock protocol data DS\_P using the above-described steps. The  
10 de-skew controller 368 includes an error counter (not shown) that counts the number of times that ERR\_Bit7 or ERR\_Bit3 is asserted during these continuing comparisons. It is noted that the sample counter included in the de-skew controller 368 also continues to count the number of  
15 times that SAM is asserted. In the event the error counter reaches the value ERR\_CMP before or at the time the sample counter reaches the value SAM\_CMP, the control signal LOCK is de-asserted and the above-described steps are repeated from the start, e.g., starting with the  
20 parallel data DS\_D15.

It will further be appreciated by those of ordinary skill in the art that modifications to and variations of the above-described system and method for transmitting parallel data over a plurality of high speed serial lines  
25 may be made without departing from the inventive concepts disclosed herein. Accordingly, the invention should not be viewed as limited except as by the scope and spirit of the appended claims.